Human-Machine Communication for Assistive IoT Technologies

Alexandra Porter Md Muztoba Umit Y. Ogras
School of Electrical, Computer, and Energy Engineering, Arizona State University

EXTENDED ABSTRACT

Despite the phenomenal advances in the computational power and functionality of electronic systems, human-machine interaction has largely been limited to simple control panels, keyboard, mouse and display. Consequently, these systems either rely critically on close human guidance or operate almost independently from the user. An exemplar technology integrated tightly into our lives is the smartphone. However, the term "smart" is a misnomer, since it has fundamentally no intelligence to understand its user. The users still have to type, touch or speak (to some extent) to express their intentions in a form accessible to the phone. Hence, intelligent decision making is still almost entirely a human task.

A life-changing experience can be achieved by transforming machines from passive tools to agents capable of understanding human physiology and what their user wants [1]. This can advance human capabilities in unimagined ways by building a symbiotic relationship to solve real world problems cooperatively. One of the high-impact application areas of this approach is assistive internet of things (IoT) technologies for physically challenged individuals. The Annual World Report on Disability reveals that 15% of the world population lives with disability, while 110 to 190 million of these people have difficulty in functioning [1]. Quality of life for this population can improve significantly if we can provide accessibility to smart devices, which provide sensory inputs and assist with everyday tasks.

This work demonstrates that smart IoT devices open up the possibility to alleviate the burden on the user by equipping everyday objects, such as a wheelchair, with decision-making capabilities. Moving part of the intelligent decision making to smart IoT objects requires a robust mechanism for human-machine communication (HMC). To address this challenge, we present examples of multimodal HMC mechanisms, where the modalities are electroencephalogram (EEG), speech commands, and motion sensing. We also introduce an IoT co-simulation framework developed using a network simulator (OMNeT++) and a robot simulation platform Virtual Robot Experimentation Platform (V-REP). We show how this framework is used to evaluate the effectiveness of different HMC strategies using automated indoor navigation as a driver application.

1. MULTIMODAL HUMAN-MACHINE COMMUNICATION

Unimodal devices such as keyboard and mouse have grown to be familiar, but they tend to restrict the information and command flow between the user and the computer system. Therefore, they are not practical in an IoT scenario. As evident from numerous studies [3][4], the interaction of humans with their environment is naturally multimodal. In order to achieve the smoothness and error tolerance of human-human interaction, we consider multimodel communication with the IoT devices [5].

CODES/ISSS '16, October 01-07, 2016, Pittsburgh, PA, USA © 2016 ACM. ISBN 978-1-4503-4483-8/16/10…$15.00
DOI: http://dx.doi.org/10.1145/2968456.2974009

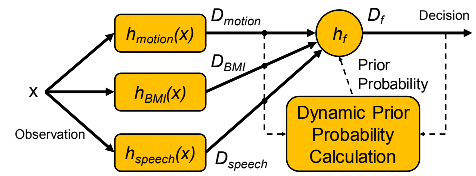

Figure 1. System architecture of the proposed multimodal HMC system.

The accuracy problem in interpreting the HMC events is a key issue in practical HMC. Fusion of multisensory data, such as EEG, speech, and motion, can be accomplished at three levels: data, feature, and decision level. Since the monitored signals are of different nature and sensed using different types of sensors, data-level fusion is not appropriate for multimodal HMC. In feature-level fusion, each stream of sensory information is first analyzed for features and then the detected features are fused. However, experimental studies show that decision level integration can improve the recognition accuracy [6]. Hence, in this work we implemented the HMC system using a decision-level fusion of multiple modalities.

We consider a multimodal HMC system consisting of a brain-machine interface (BMI), a speech recognizer, and a motion detector, as shown in Figure 1. Each classifier $\{h_{motion}, h_{BMI}, h_{speech}\}$ calculates the features from an observation, x, compares the features and makes a decision. For example, consider a user intent of steering *Left* while navigating a power wheelchair. The motion detector captures the accelerometer and gyroscope sensor data from the observation, Left Gesture. The classifier, h_{motion}, calculates the features, such as roll, pitch, and yaw, compares the feature values and reaches a decision, D_{motion}, which can be one of the events supported by the motion detector, such as forward and left gestures. All the supported events across all classifiers map to commands required in the navigation application, for example, {F, R, L, S, …}. Then, the proposed fusion classifier, h_f, calculates the credibility of the decisions $\{D_{BMI}, D_{motion}, D_{speech}\}$ from the prior probabilities of correct interpretation. We use these values as weights to determine the fused decision D_f, as shown in Figure 1. Evaluating the effectiveness of the proposed HMC system requires modeling of the communication as well as the physical world. The following section discusses the co-simulation framework used for evaluation.

2. IOT CO-SIMULATION FRAMEWORK

The proposed IoT co-simulation framework consists of three interconnected layers, as shown in Figure 2.

1. **Physical Layer** for modeling the physical world,
2. **Control Layer** for modeling the behavior of the objects in the physical layer,
3. **Network Layer** for modeling the communication network.

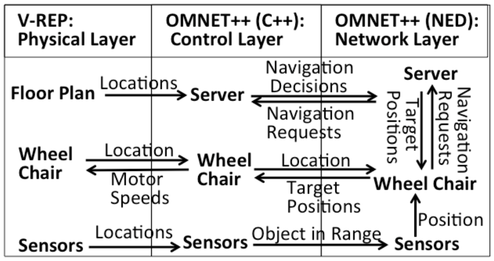

Figure 2. Structure of the proposed IoT co-simulation framework.

The physical layer is implemented in V-REP [7]. To model a realistic building in V-REP, we first place physical entities, such as walls, doors, and sensors, in a scene. Then, we add the human models and the IoT objects, such as a wheelchair, to the scene. V-REP accurately models the movements of the human models and wheelchair in the 3-dimensional space.

The control and network layers are implemented using OMNeT++, a C++ simulation library and network simulation framework [8]. Communication between V-REP and OMNeT++ is enabled through the application programming interface (API) provided by V-REP. In OMNeT++, entities, such as the wheelchair, sensors, doors, servers, or a central controller, are each represented by C++ classes and a network description file (NED). The C++ classes correspond to the control layer of the simulation, representing each object's response to network messages. When responses include physical motion, such as controlling the wheelchair, the C++ class also acts on the V-REP model, such as by triggering the corresponding V-REP object to set the motor speeds. The network layer is represented by the NED files, which specify the wireless protocols being used to send packages, physical locations of sensors, and variables on the capabilities of the network, such as speed and capacity.

2.1 Overview of the Operation

To illustrate the dynamics of the simulation, we use assisted indoor navigation as a driver application. The floor plan, containing positions of all the objects in the V-REP model, including the initial positions of the human models and the wheelchair, is passed from VREP to OMNeT++ in the beginning of the simulation. The wheelchair model receives the navigation commands from the user. This interaction employs multimodal communication, as explained in Section 1. In our setup, a BMI headset and a motion sensor pack send the user commands to the host computer using Bluetooth LE. The C++ class that models the wheelchair uses the fused command, along with its current position and orientation, to compute the target velocity of the wheelchair. The target velocity is then used to compute the speed of the right and left wheels of the wheelchair using a kinematic model. V-REP takes these inputs and moves the wheelchair accordingly. The control classes in OMNeT++ requests the V-REP representation's position at regular intervals (1s by default) to make course adjustments. Sensor proximity generates additional feedback on the position. It is assumed the wheelchair controller would not be able to accurately gauge the position of the chair as it moves in a realistic setting, so localization error is randomly accumulated on the wheelchair positions stored in the C++ class.

Communication between objects is simulated in OMNeT++. For example, if the wheelchair is in the vicinity of an RFID used for global positioning, the broadcast message from the RFID is transmitted following the wireless communication protocol specified in the simulation setup. Similarly, if two distant object need to communicate, the messages are transmitted through a multi-hop ad hoc network modeled using OMNeT++.

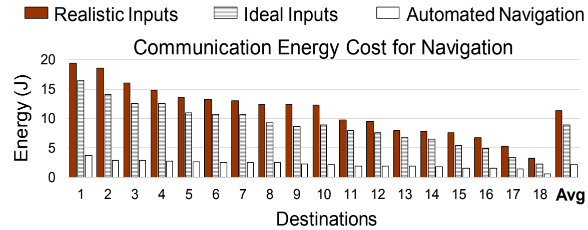

Figure 3. Simulation result for communication energy consumption when realistic, ideal inputs, and automated navigation using realistic inputs are used in navigation.

2.2 Sample Results

The proposed co-simulation framework enables evaluation of complex IoT application by considering the tight interactions among different layers. Hence, it eliminates the design gap due to the late integration of different layers. Our framework calculates parameters, such as positioning error, navigation time, user effort, communication energy and false event generations. The type of results that can be obtained using the proposed co-simulation framework are illustrated in Figure 3. In this experiment, we considered three different scenarios in which a wheelchair is navigated to 18 different destinations in a virtual home. First, ideal user inputs are used to control a wheelchair. Then, realistic inputs derived from user gestures and BMI are used to control the wheelchair. Ideal inputs allow perfect control of the wheelchair and hence reduced *communication energy* than realistic inputs. Finally, we simulated an automated indoor navigation algorithm that uses minimal number of realistic inputs from the user. The automated navigation saves significant amount of *communication* energy by reducing number of command transmitted from the user. That is, moving part of the intelligent decision making to the object under control reduces the communication energy.

In summary, design choices in the physical, control, and network layer affect the overall performance. Our co-simulation framework enables optimization at each layer thorough exhaustive experimentation, leading to a better understanding of how the entire system interacts.

3. REFERENCES

[1] Babak Mahmoudi and Justin C Sanchez. "A Symbiotic Brain-Machine Interface through Value-based Decision Making." PloS one, 6(3):e14760, 2011.

[2] World Health Organization. Annual World Report on Disability, Available Online (last accessed 4 July 2016). http://www.who.int/disabilities/world_report/2011/en/

[3] Sharma, Rajeev, Vladimir I. Pavlovic, and Thomas S. Huang. "Toward Multimodal Human-Computer Interface." Proceedings of the IEEE 86.5 (1998): 853-869.

[4] Jaimes, Alejandro, and Nicu Sebe. "Multimodal Human–Computer Interaction: A survey." Computer Vision and Image Understanding 108.1 (2007): 116-134.

[5] Pantic, Maja, and Leon JM Rothkrantz. "Toward an Affect-Sensitive Multimodal Human-Computer Interaction." Proceedings of the IEEE 91.9 (2003): 1370-1390.

[6] P. Scanlon and R. B. Reilly, "Feature Analysis for Automatic Speech Reading," in Proc. IEEE Int. Workshop Multimedia Signal Processing, 2001, pp. 625–630.

[7] V-REP - Virtual Robot Experimentation Platform, [Online] http://www.coppeliarobotics.com/, Accessed 10 July 2016.

[8] OMNeT++ Discrete Event Simulator, [Online] https://omnetpp.org/, Accessed 10 July 2016